

# GWT From Scratch – Day 2

## Playing With Widgets

The aim of today's session is to familiarise you with some GWT widgets and how to get them onto your screen.

### **We will look at**

- Labels
- Slots in your HTML page
- The TextBox
- The Button
- Adding a ClickListener to a button
- The VerticalPanel
- The CheckBox
- The HTML Widget
- RadioButtons
- TextAreas
- ToggleButtons
- Some CSS to add a bit of style
- Finally we will set up a SuggestBox which will autosuggest all the countries in the world

### **Questions?**

My ideal is to have no questions at all because everything is perfectly clear, but if you have any questions, then please contact me. I would like to consider the course 'complete' within the limits I have set for it. In other words, I am here to supplement the course if, in any way, it is not well enough explained to get everyone through it. I intend to use the feedback to improve the course and reduce the questions. So any questions are very welcome.

### **Any Problems**

[rx01-day2@examples.roughian.com](mailto:rx01-day2@examples.roughian.com)

© Ian Bambury 2008

## Playing With Widgets

### The Label

Yes, I know, we've already got that. But we going to change it a little bit so that we don't lose it every time we look at a new widget.

Cut and paste the following code into the Main.java file, replacing the previous contents.

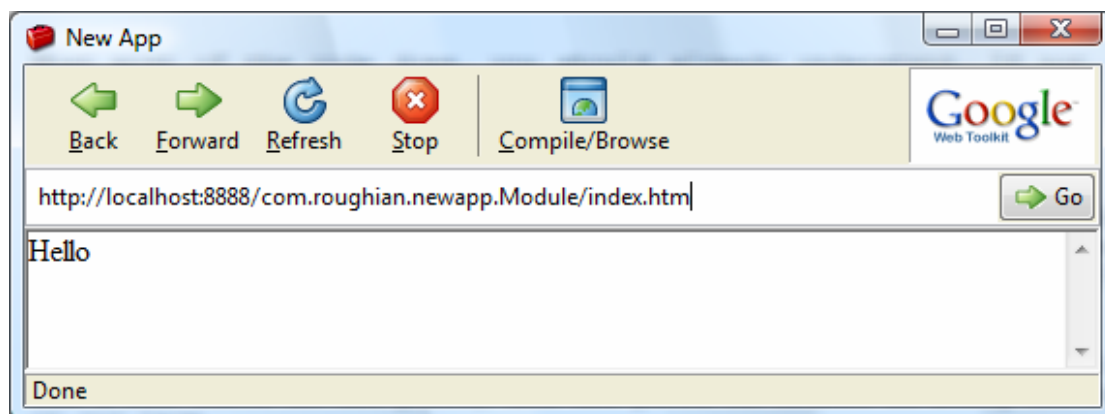
```
package com.roughian.newapp.client;
import com.google.gwt.core.client.EntryPoint;
import com.google.gwt.user.client.ui.Label;
import com.google.gwt.user.client.ui.RootPanel;
import com.google.gwt.user.client.ui.VerticalPanel;
public class Main implements EntryPoint
{
    VerticalPanel panel = new VerticalPanel();
    public void onModuleLoad()
    {
        RootPanel.get("slot").add(panel);
        panel.add(new Label("Hello"));
    }
}
```

What most of the code does, you should already understand. If not, review the previous session.

The new feature in this bit of code is the VerticalPanel.

### The VerticalPanel

The VerticalPanel is a table-based widget which is one cell wide and adds cells downwards as you add widgets to it. It is declared at the class level because we need to refer to it from within a number of methods.



In the `onModuleLoad()` method we add the panel to the `RootPanel` and we add a new label to the panel, and the panel of course is being shown on our page.

If you run the code by clicking the run button you will see exactly the same as before - the word 'Hello' should appear on the screen.

## A Bit More About The Slot

You could add the panel directly to the `RootPanel` by not specifying the slot ID in the 'get'. In this particular case you would get the same result, i.e. the word 'Hello'.

The advantage of slots when things become more complicated is that you can define exactly where you want things to go.

In practice, I have found for fairly simple sites that the best approach is to have a single slot in your HTML and add a `Site` class to it. It's a lot easier to control everything within the Java. If you're working alone then this is a good solution, and since you are learning GWT at the moment, we'll stick with that solution.

In this simple example then, our `VerticalPanel` is effectively our site.

## The TextBox

Copy the 'panel.add' line and paste in a clone of it underneath. In the new line, change

```
Label("Hello") to TextBox()
```

Noted that Eclipse has put a squiggly underline under it and also flagged it with a light bulb and a red circle with a white cross on the left-hand side in the gutter.

If you put your mouse pointer over either of these error indicators then you'll get a message saying that the `TextBox` cannot be resolved to a type. What this means is that Eclipse can't find the `TextBox` class.

What we need to resolve this situation is an import. Rather than typing it in, place the cursor at the end of the word 'TextBox' and press `Control+SpaceBar`. This will give you a list of everything that Eclipse can find which might be a match for that word.

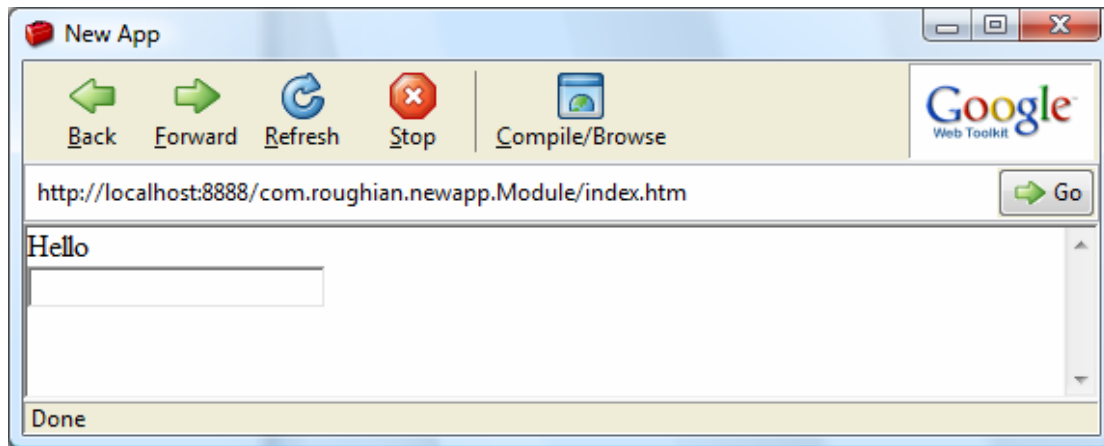
Choose the line

```
TextBox - com.google.gwt.user.client.ui
```

which will probably be the first one. If it is and is already highlighted, then just press enter, otherwise you have to select it with the arrow keys on your mouse and press enter, or double click it.

Save your changes - you can use `Control+S` to save just this file (since we only have one). Personally I have got into the habit of pressing `Control+Shift+S` which saves all open files.

Now go back to your hosted browser window and click the Refresh button. You should see the `TextBox` appear as well as the word 'Hello'.



## Using The TextBox Properties

A TextBox like this is not much use. It is 'anonymous', i.e. it doesn't have a name.

Quite often widgets don't need a name. Static text in labels for example, you will never need to refer to. The same goes for buttons. For anything where you will want to retrieve the contents or what has been selected, you will almost certainly need to name it - at least, you will in simple situations.

We need to change our little program then, so we can get at the widgets and their properties. Paste over the TextBox line with the following code:

```
TextBox textBox = new TextBox();  
panel.add(textBox);
```

This allows us to get at the TextBox.

Underneath this code, add the following code:

```
textBox.setText("This is a text box");
```

This, as you might have guessed, adds some text to the TextBox. Don't worry, things are going to speed up in a minute.

Give it a try, click the Refresh button in the hosted mode browser.

## Adding A Button

Replace all the code in the file with the following:

```
package com.roughian.newapp.client;  
  
import com.google.gwt.core.client.EntryPoint;  
import com.google.gwt.user.client.ui.Button;  
import com.google.gwt.user.client.ui.ClickListener;  
import com.google.gwt.user.client.ui.Label;  
import com.google.gwt.user.client.ui.RootPanel;  
import com.google.gwt.user.client.ui.TextBox;
```

```

import com.google.gwt.user.client.ui.VerticalPanel;
import com.google.gwt.user.client.ui.Widget;

public class Main implements EntryPoint
{
    final static String NOT_SET = "[Not Set]";
    VerticalPanel panel = new VerticalPanel();
    Button button = new Button("Transfer");
    TextBox textBox = new TextBox();
    Label label = new Label(NOT_SET);

    public void onModuleLoad()
    {
        RootPanel.get("slot").add(panel);
        panel.add(textBox);
        panel.add(button);
        panel.add(label);

        button.addClickListener(listener);
    }
    ClickListener listener = new ClickListener(){
        public void onClick(Widget sender)
        {
            String text = textBox.getText();
            int length = text.length();
            if(length == 0) text = NOT_SET;
            label.setText(text);
            textBox.setText("");
        }
    };
}

```

This is properly not the best way to code this program. But it is easier to understand like this, and anyway, the program doesn't actually do anything useful apart from illustrate some things for me.

Running down the program, then:

Package name you know about.

Imports you know about.

'final static String' is basically just a constant. It's a good idea to get into the habit of using constants if you can.

The VerticalPanel

Next we have VerticalPanel which contains all our other bits.

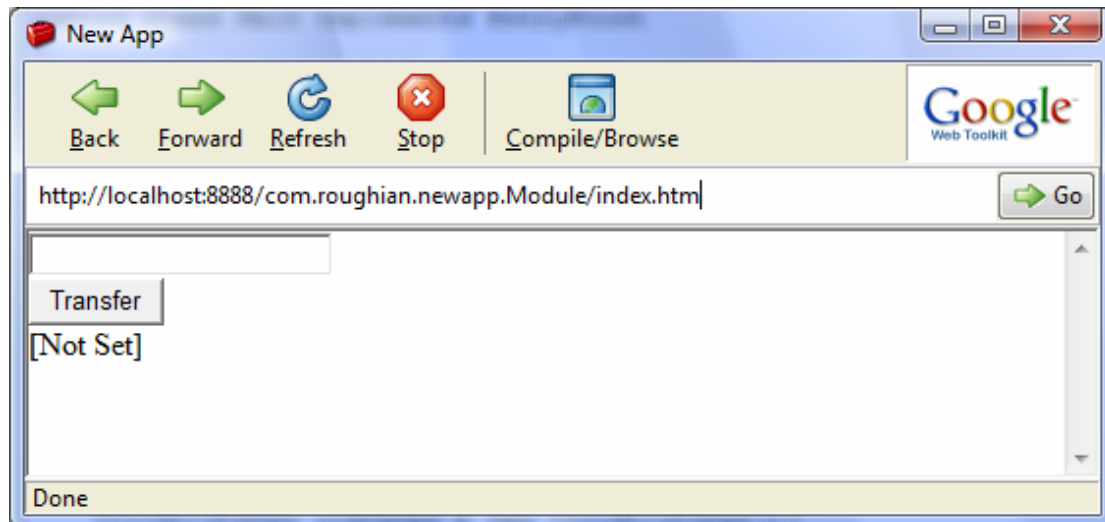
Then we have the widgets which are going to go onto the page, declared at class level so that we can get at them from anywhere.

In the onModuleLoad(), we add the panel to the screen, and all our bits and pieces to the panel. And in widgets to the screen as soon as possible is a good habit to get into. Here it would make any difference, but there are some things, especially getting

positions and heights and widths that depend on the widget being attached to the DOM.

In the last line of the `onModuleLoad()` method, we had a `ClickListener` to our button. After that, we have a `ClickListener` itself.

In the event of someone clicking the button, the code in the `onClick()` method will run. In a rather verbose way, we get the text from the `TextBox`, we get the length of the text, if nothing has been entered then we set our string called 'text' to "[Not Set]", we set the text of the label, and then we clear the text out of the `TextBox`.



In tomorrow's session, we'll do more on listeners in general, all the different kinds (well a lot of them, anyway, enough so you'll be to sort out the others yourself), and the different ways you can define and use them. But for now we'll have a look at some more widgets.

## More Simple Widgets

Try pasting the following code at the bottom of the `onModuleLoad()` method:

```
CheckBox checkBox = new CheckBox("CheckBox label");
panel.add(checkBox);
HTML html = new HTML("Here is <i><b>some</b></i> text");
panel.add(html);
panel.add(new RadioButton("group", "One"));
panel.add(new RadioButton("group", "Two"));
panel.add(new RadioButton("group", "Three"));
TextArea textArea = new TextArea();
panel.add(textArea);

ToggleButton toggleButton = new ToggleButton("Click Me",
        "And Again To Release");
panel.add(toggleButton);
```

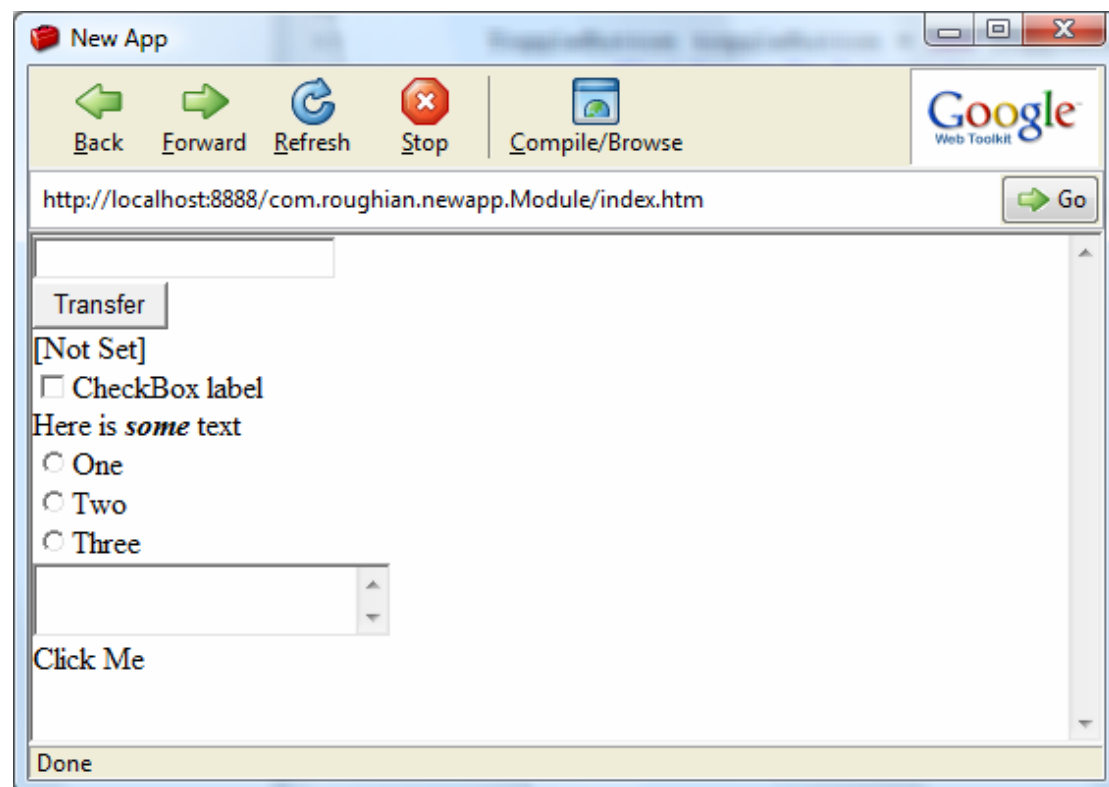
You'll find an awful lot of squiggly lines.

A quick way out of this (in Eclipse) is to head up to the 'import' area, pick on of the existing ui lines (say, the Label one) and change (in this case) 'Label' to '\*'. This includes everything in the package, which will do, but is not ideal.

So with Control and Shift both kept pressed, press the 'o' key, the 'f' key and the 's' key. This 'o'rganises the imports, 'f'ormats the file, and 's'aves all open files.

I use this sequence a lot - almost instinctively when I am about to run the program.

Run it in the usual way and you'll see a mess of widgets on the screen. We're going to need some formatting, aren't we? But unfortunately that comes later. At the moment we're just concentrating on widgets and what they can do.



## GWT's Got No Style

GWT widgets don't come with any style at all in most cases. Have a look at the ToggleButton. Mouse over it and you'll get the I-beam, there is no border on the button at all, and nothing to indicate that you can click it. But if you click it, it still works.

One very useful method when you come to layouts is the 'setBorderWidth()'. Try putting this line after the RootPanel line:

```
panel.setBorderWidth(5);
```

Give it a whirl, and have a look at the results. It's not something you would want to use in your final application because it is as ugly as sin, but it can be extremely useful when you're trying to work out where everything has gone.

## **Being Suggestive**

Last one for today. We'll set up a suggestion box, and will even give it a bit of style.

Replace everything in Main.java with the following:

```
package com.roughian.newapp.client;

import com.google.gwt.core.client.EntryPoint;
import com.google.gwt.user.client.ui.Label;
import com.google.gwt.user.client.ui.MultiWordSuggestOracle;
import com.google.gwt.user.client.ui.RootPanel;
import com.google.gwt.user.client.ui.SuggestBox;
import com.google.gwt.user.client.ui.VerticalPanel;

public class Main implements EntryPoint
{
    public void onModuleLoad()
    {
        demo();
    }

    public void demo()
    {
        VerticalPanel panel = new VerticalPanel();
        SuggestBox suggestbox = new
            SuggestBox(createCountriesOracle());
        panel.add(new Label("Enter Country"));
        panel.add(suggestbox);
        RootPanel.get("slot").add(panel);
    }

    MultiWordSuggestOracle createCountriesOracle()
    {
        MultiWordSuggestOracle oracle = new
MultiWordSuggestOracle();
        oracle.add("Afghanistan");
        oracle.add("Albania");
        oracle.add("Algeria");
        oracle.add("American Samoa");
        oracle.add("Andorra");
        oracle.add("Angola");
        oracle.add("Anguilla");
        oracle.add("Antarctica");
        oracle.add("Antigua And Barbuda");
        oracle.add("Argentina");
        oracle.add("Armenia");
        oracle.add("Aruba");
        oracle.add("Australia");
        oracle.add("Austria");
        oracle.add("Azerbaijan");
        oracle.add("Bahamas");
        oracle.add("Bahrain");
        oracle.add("Bangladesh");
        oracle.add("Barbados");
        oracle.add("Belarus");
    }
}
```



```
oracle.add("Belgium");
oracle.add("Belize");
oracle.add("Benin");
oracle.add("Bermuda");
oracle.add("Bhutan");
oracle.add("Bolivia");
oracle.add("Bosnia And Herzegovina");
oracle.add("Botswana");
oracle.add("Bouvet Island");
oracle.add("Brazil");
oracle.add("British Indian Ocean Territory");
oracle.add("Brunei Darussalam");
oracle.add("Bulgaria");
oracle.add("Burkina Faso");
oracle.add("Burundi");
oracle.add("Cambodia");
oracle.add("Cameroon");
oracle.add("Canada");
oracle.add("Cape Verde");
oracle.add("Cayman Islands");
oracle.add("Central African Republic");
oracle.add("Chad");
oracle.add("Chile");
oracle.add("China");
oracle.add("Christmas Island");
oracle.add("Cocos (Keeling) Islands");
oracle.add("Colombia");
oracle.add("Comoros");
oracle.add("Congo, The Democratic Republic Of The");
oracle.add("Congo");
oracle.add("Cook Islands");
oracle.add("Costa Rica");
oracle.add("Cote D'Ivoire");
oracle.add("Croatia");
oracle.add("Cuba");
oracle.add("Cyprus");
oracle.add("Czech Republic");
oracle.add("Denmark");
oracle.add("Djibouti");
oracle.add("Dominica");
oracle.add("Dominican Republic");
oracle.add("East Timor");
oracle.add("Ecuador");
oracle.add("Egypt");
oracle.add("El Salvador");
oracle.add("Equatorial Guinea");
oracle.add("Eritrea");
oracle.add("Estonia");
oracle.add("Ethiopia");
oracle.add("Falkland Islands (Malvinas)");
oracle.add("Faroe Islands");
oracle.add("Fiji");
oracle.add("Finland");
oracle.add("France");
oracle.add("French Guiana");
```

```
oracle.add("French Polynesia");
oracle.add("French Southern Territories");
oracle.add("Gabon");
oracle.add("Gambia");
oracle.add("Georgia");
oracle.add("Germany");
oracle.add("Ghana");
oracle.add("Gibraltar");
oracle.add("Greece");
oracle.add("Greenland");
oracle.add("Grenada");
oracle.add("Guadeloupe");
oracle.add("Guam");
oracle.add("Guatemala");
oracle.add("Guinea-Bissau");
oracle.add("Guinea");
oracle.add("Guyana");
oracle.add("Haiti");
oracle.add("Heard Island And Mcdonald Islands");
oracle.add("Holy See (Vatican City State)");
oracle.add("Honduras");
oracle.add("Hong Kong");
oracle.add("Hungary");
oracle.add("Iceland");
oracle.add("India");
oracle.add("Indonesia");
oracle.add("Iran, Islamic Republic Of");
oracle.add("Iraq");
oracle.add("Ireland");
oracle.add("Israel");
oracle.add("Italy");
oracle.add("Jamaica");
oracle.add("Japan");
oracle.add("Jordan");
oracle.add("Kazakstan");
oracle.add("Kenya");
oracle.add("Kiribati");
oracle.add("Korea, Democratic People's Republic Of");
oracle.add("Korea, Republic Of");
oracle.add("Kuwait");
oracle.add("Kyrgyzstan");
oracle.add("Lao People's Democratic Republic");
oracle.add("Latvia");
oracle.add("Lebanon");
oracle.add("Lesotho");
oracle.add("Liberia");
oracle.add("Libyan Arab Jamahiriya");
oracle.add("Liechtenstein");
oracle.add("Lithuania");
oracle.add("Luxembourg");
oracle.add("Macau");
oracle.add("Macedonia, The Former Yugoslav Republic
Of");
oracle.add("Madagascar");
oracle.add("Malawi");
```

```
oracle.add("Malaysia");
oracle.add("Maldives");
oracle.add("Mali");
oracle.add("Malta");
oracle.add("Marshall Islands");
oracle.add("Martinique");
oracle.add("Mauritania");
oracle.add("Mauritius");
oracle.add("Mayotte");
oracle.add("Mexico");
oracle.add("Micronesia, Federated States Of");
oracle.add("Moldova, Republic Of");
oracle.add("Monaco");
oracle.add("Mongolia");
oracle.add("Montserrat");
oracle.add("Morocco");
oracle.add("Mozambique");
oracle.add("Myanmar");
oracle.add("Namibia");
oracle.add("Nauru");
oracle.add("Nepal");
oracle.add("Netherlands Antilles");
oracle.add("Netherlands");
oracle.add("New Caledonia");
oracle.add("New Zealand");
oracle.add("Nicaragua");
oracle.add("Niger");
oracle.add("Nigeria");
oracle.add("Niue");
oracle.add("Norfolk Island");
oracle.add("Northern Mariana Islands");
oracle.add("Norway");
oracle.add("Oman");
oracle.add("Pakistan");
oracle.add("Palau");
oracle.add("Palestinian Territory, Occupied");
oracle.add("Panama");
oracle.add("Papua New Guinea");
oracle.add("Paraguay");
oracle.add("Peru");
oracle.add("Philippines");
oracle.add("Pitcairn");
oracle.add("Poland");
oracle.add("Portugal");
oracle.add("Puerto Rico");
oracle.add("Qatar");
oracle.add("Reunion");
oracle.add("Romania");
oracle.add("Russian Federation");
oracle.add("Rwanda");
oracle.add("Saint Helena");
oracle.add("Saint Kitts And Nevis");
oracle.add("Saint Lucia");
oracle.add("Saint Pierre And Miquelon");
oracle.add("Saint Vincent And The Grenadines");
```

```
oracle.add("Samoa");
oracle.add("San Marino");
oracle.add("Sao Tome And Principe");
oracle.add("Saudi Arabia");
oracle.add("Senegal");
oracle.add("Seychelles");
oracle.add("Sierra Leone");
oracle.add("Singapore");
oracle.add("Slovakia");
oracle.add("Slovenia");
oracle.add("Solomon Islands");
oracle.add("Somalia");
oracle.add("South Africa");
oracle.add("South Georgia And The South Sandwich
Islands");
oracle.add("Spain");
oracle.add("Sri Lanka");
oracle.add("Sudan");
oracle.add("Suriname");
oracle.add("Svalbard And Jan Mayen");
oracle.add("Swaziland");
oracle.add("Sweden");
oracle.add("Switzerland");
oracle.add("Syrian Arab Republic");
oracle.add("Taiwan, Province Of China");
oracle.add("Tajikistan");
oracle.add("Tanzania, United Republic Of");
oracle.add("Thailand");
oracle.add("Togo");
oracle.add("Tokelau");
oracle.add("Tonga");
oracle.add("Trinidad And Tobago");
oracle.add("Tunisia");
oracle.add("Turkey");
oracle.add("Turkmenistan");
oracle.add("Turks And Caicos Islands");
oracle.add("Tuvalu");
oracle.add("Uganda");
oracle.add("Ukraine");
oracle.add("United Arab Emirates");
oracle.add("United Kingdom");
oracle.add("United States Minor Outlying Islands");
oracle.add("United States");
oracle.add("Uruguay");
oracle.add("Uzbekistan");
oracle.add("Vanuatu");
oracle.add("Venezuela");
oracle.add("Viet Nam");
oracle.add("Virgin Islands, British");
oracle.add("Virgin Islands, U.S.");
oracle.add("Wallis And Futuna");
oracle.add("Western Sahara");
oracle.add("Yemen");
oracle.add("Yugoslavia");
oracle.add("Zambia");
```

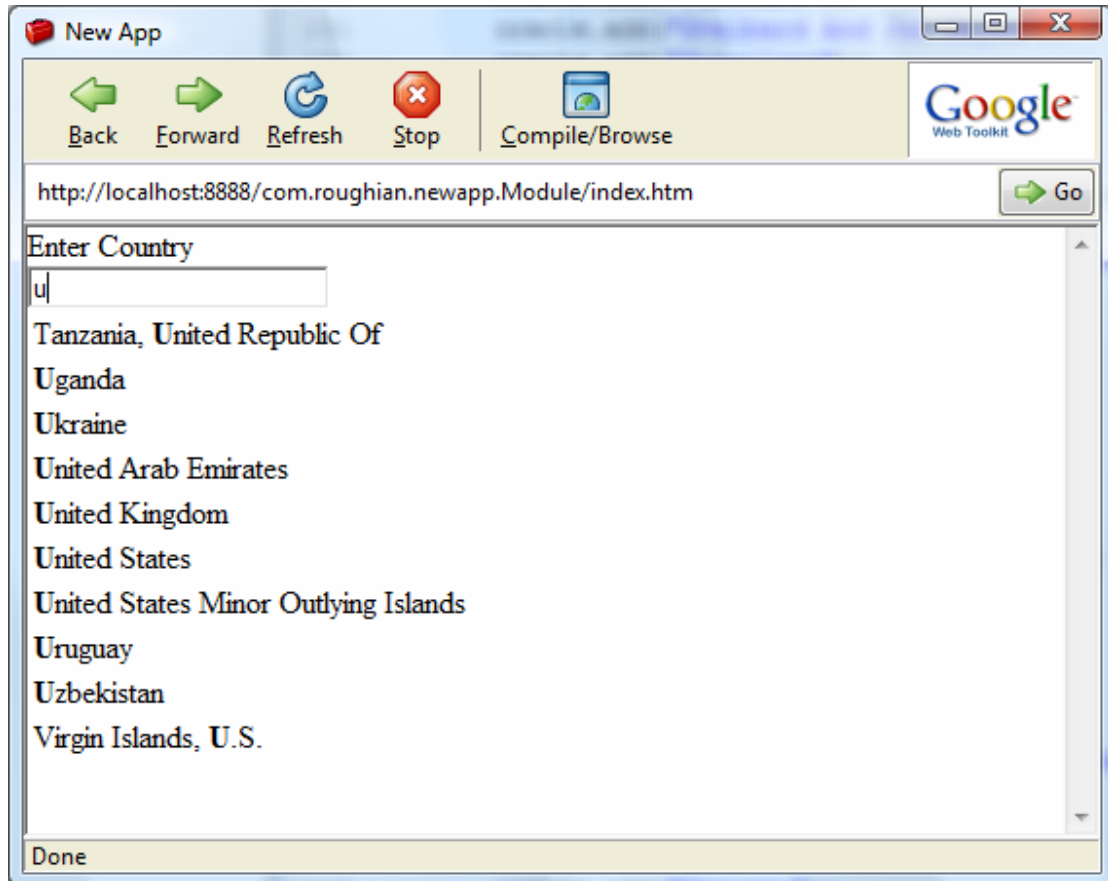
```

        oracle.add( "Zimbabwe" );

        return oracle;
    }
}

```

Give it a try and see what happens. Type in a 'u' and you will get all the countries that start with that letter or have words in their name that start with that letter.



If you click on one of the country names, then it will be entered into the TextBox. But it doesn't look like it will, does it? I think it's worth actually having a look at this piece of programming without the CSS so you can see the difference when we add it.

Change to the index.htm file, and paste the following CSS between the style tags:

```

body
{
    font-family                :    sans-serif;
}
table
{
    border-collapse            :    collapse;
}
.gwt-SuggestBox
{
    border                    :    1px solid #666;
    text-align                :    left;
}

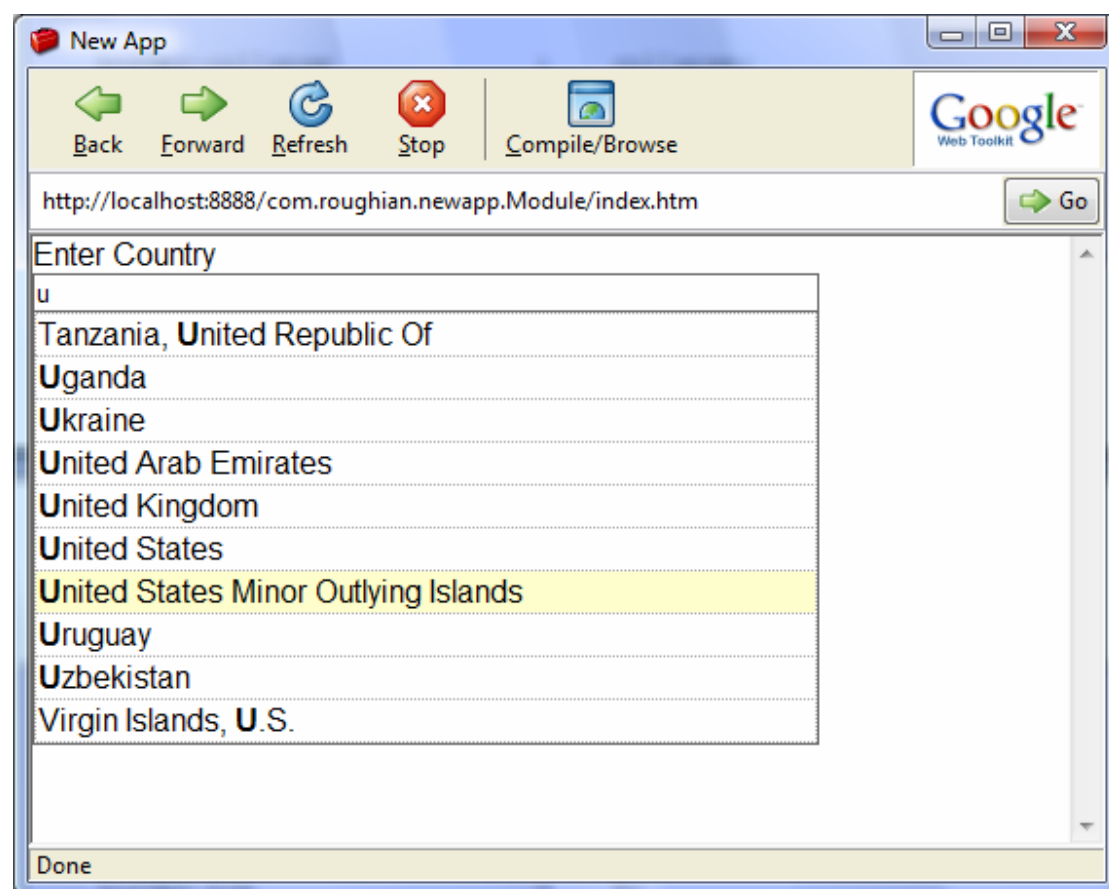
```

```

width                :    400px;
}
.gwt-SuggestBoxPopup
{
    text-align        :    left;
    cursor            :    pointer;
    cursor            :    hand;
    border             :    1px solid #666;
    border-top        :    0;
    background-color   :    #fff;
}
.gwt-SuggestBoxPopup .item
{
    border             :    1px dotted #aaa;
    width              :    398px;
}
.gwt-SuggestBoxPopup .item-selected
{
    background-color   :    #ffc;
}

```

Try it out. It makes quite a difference, doesn't it?



Obviously in a real situation, you are unlikely to want to hard-code everything in the program, but here, if we didn't, we'd have to go and get it from a database, and that's not going to be covered today.

## **Next Time**

Tomorrow, as I said, we'll be looking at listeners.

If you want more on widgets then you can have a look at any of the pages at

<http://examples.roughian.com/#Widgets~Summary>